Information and Computation 208 (2010) 982–995

Contents lists available at ScienceDirect

Information and Computation

journal homepage: www.elsevier.com/locate/ic

On probabilistic pushdown automata ☆

Juraj Hromkovič [a,*,1], Georg Schnitger [b,2]

[a] Departement Informatik, ETH Zürich, CAB F16, Universitätsstrasse 6, 8092 Zürich, Switzerland
[b] Institut für Informatik, Johann Wolfgang Goethe-Universität, Robert Mayer Straße 11–15, 60054 Frankfurt am Main, Germany

ARTICLE INFO

Article history:
Received 10 March 2009
Revised 4 November 2009
Available online 23 November 2009

This paper is dedicated to the remembrance of Rainer Kemp.

Keywords:
Pushdown automata
Determinism
Nondeterminism
Randomization

ABSTRACT

We study the most important probabilistic computation modes for pushdown automata. First we show that deterministic pushdown automata (pda) are weaker than Las Vegas pda, which in turn are weaker than one-sided-error pda. Next one-sided-error pda are shown to be weaker than (nondeterministic) pda. Finally bounded-error two-sided error pda and nondeterministic pda are incomparable. To show the limited power of bounded-error two-sided pda we apply communication arguments; in particular we introduce a non-standard model of communication which we analyze with the help of the discrepancy method.

The power of randomization for pda is considerable, since we construct languages which are not deterministic context-free (resp. not context-free) but are recognizable with even arbitrarily small error by one-sided-error (resp. bounded-error) pda. On the other hand we show that, in contrast to many other fundamental models of computing, error probabilities can in general not be decreased arbitrarily: we construct languages which are recognizable by one-sided-error pda with error probability $\frac{1}{2}$, but not by one-sided-error pushdown automata with error probability $p < \frac{1}{2}$. A similar result, with error probability $\frac{1}{3}$, holds for bounded-error two-sided error pda.

1. Introduction

Although randomization is by now a standard tool for making computations more efficient or for building simpler systems, we are far from fully understanding the power of randomized computing. Hence it is advisable to study randomization for restricted models of computation. This line of research has started with the study of simple models like one-way finite automata and two-party communication protocols and continues with the investigation of more and more complex models of computation (see [1,2,4,6–9,13,14,17] among others). We follow this approach by investigating the power of randomization for pushdown automata.

Pushdown automata (pda) are one of the classical models of computation presented in each theoretical computer science textbook, since nondeterministic pushdown automata (npda) define the well-known class of context-free languages (*CF*) and deterministic pushdown automata (dpda) define the class of deterministic context-free languages (*DCF*). In contrast to the intensive investigation of different versions of probabilistic finite automata, very little is known about probabilistic pda. Freivalds [3] shows that probabilistic pda even with arbitrarily small error probability recognize more languages than dpda. In [13] it is shown that there is no difference between determinism, nondeterminism and bounded-error randomness for

☆ This paper is based on publications [10,11].

* Corresponding author.
Email addresses: juraj.hromkovic@inf.ethz.ch (J. Hromkovič), georg@thi.informatik.uni-frankfurt.de (G. Schnitger).

[1] Partially supported by SNF grant 200020-120073/1.
[2] Partially supported by DFG grant SCHN 503/4-1.

0890-5401/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ic.2009.11.001

pushdown automata recognizing tally languages. Further results are known for unbounded-error randomization [15], but these results are not applicable to our bounded-error setting.

Definition 1. We define a probabilistic pda P as a nondeterministic pda with a probability distribution over the next moves and demand that all computations are finite. We say that P recognizes a language L with error at most $\varepsilon()$, iff for each $x \in L$, prob[P accepts x] $\geq 1 - \varepsilon(|x|)$ and for each $x \notin L$, prob[A rejects x] $\geq 1 - \varepsilon(|x|)$. We demand that all computations of P are finite.

We next give a brief introduction to the different modes of probabilistic pda considered in this paper. In particular we emphasize probabilistic pda with error amplification (i.e., decreasing error probability arbitrarily), since this model provides a natural extension of dpda's and hence of deterministic context-free languages.

The states of a Las Vegas pda are partitioned into the sets of accepting, rejecting and neutral states; ε-moves from a state in one of the three classes to a state in a different class are not allowed. A Las Vegas pda is not forced to give a definite answer, but may instead reply with "I don't know" (when reaching a neutral state). Of course the probability of giving a non-committal answer should be as small as possible.

A Las Vegas pda A for a language L is not allowed to err, i.e., no computation rejects a word in L and no computation accepts a word from the complement of L. Formally, we say that a Las Vegas pda A recognizes L with probability at least $1 - \epsilon$, $0 \leq \epsilon < 1$, if A never errs, and if the probability of reaching a neutral state is bounded by ϵ for every input. LVCF_ϵ denotes the set of languages recognized by Las Vegas pushdown automata with probability at least $1 - \epsilon$. We set

$$\text{LVCF} = \bigcup_{0<\epsilon<1} \text{LVCF}_\epsilon \quad \text{and} \quad \text{LVCF}^* = \bigcap_{0<\epsilon<1} \text{LVCF}_\epsilon.$$

Thus LVCF consists of all languages recognizable by Las Vegas pda's, where the probability of the "I don't know" answer is separated away from 1, but may be arbitrarily large. LVCF^* is defined similarly, but now the probability of the "I don't know" answer has to be made arbitrarily small.

When considering one-sided and two-sided error randomization we again assume that there is no ϵ-move from an accepting state to a rejecting state and vice versa. In contrast to a Las Vegas automaton a one-sided or two-sided error pda has only accepting and rejecting states.

We say that a probabilistic pda A is a one-sided-error pda that recognizes a language $L(A)$ with error probability at most ϵ iff

(i) for every $w \in L(A)$, $\Pr(A \text{ accepts } w) \geq 1 - \epsilon$, and
(ii) for every $w \notin L(A)$, $\Pr(A \text{ rejects } w) = 1$.

We define RandomCF_ϵ to be the set of languages recognized by one-sided error pushdown automata with error probability at most ϵ and introduce the classes

$$\text{RandomCF} = \bigcup_{0<\epsilon<1} \text{RandomCF}_\epsilon, \quad \text{RandomCF}^* = \bigcap_{0<\epsilon<1} \text{RandomCF}_\epsilon.$$

Finally we say that a probabilistic pda A is a two-sided error pda that recognizes $L(A)$ with error probability at most ϵ iff

(i) for every $w \in L(A)$, $\Pr(A \text{ rejects } w) \leq \epsilon$, and
(ii) for every $w \notin L(A)$, $\Pr(A \text{ accepts } w) \leq \epsilon$.

The set of languages recognized by two-sided error pda's with error probability at most ϵ will be denoted by BPCF_ϵ and we also introduce

$$\text{BPCF} = \bigcup_{0<\epsilon<1/2} \text{BPCF}_\epsilon, \quad \text{BPCF}^* = \bigcap_{0<\epsilon<1/2} \text{BPCF}_\epsilon.$$

Two-sided error pda's are very powerful. It's not hard to show that BPCF is closed under complementation, under finite union and consequently under finite intersection. Thus BPCF contains languages outside of CF, since DCF is contained in BPCF.

In our main result we separate almost all the classes we just introduced.

Theorem 2.

(*a*) $\text{DCF} \subseteq \text{LVCF} \subseteq \text{RandomCF} \subseteq \text{CF}$ *as well as* $\text{RandomCF} \subseteq \text{BPCF}$ *and all inclusions are proper*.
(*b*) CF *and* BPCF *are incomparable*.
(*c*) *All inclusions* $\text{LVCF}^* \subseteq \text{LVCF}$, $\text{RandomCF}^* \subseteq \text{RandomCF}$ *and* $\text{BPCF}^* \subseteq \text{BPCF}$ *are proper*.
(*d*) $\text{LVCF}^* \subseteq \text{RandomCF}^* \subseteq \text{BPCF}^*$ *and all inclusions are proper*.
(*e*) *The following pairs of classes are incomparable*:

- RandomCF^* *and* LVCF,
- BPCF^* *and* LVCF,

- BPCF* *and* RandomCF,
- BPCF* *and* CF.

Observe that Theorem 2 gives, with one exception, a complete characterization of all inter-class relations. However the question of whether DCF is a proper subset of LVCF* is left open.

The proof of Theorem 2 requires four basic separation results. First we show that randomization remains powerful even if error probabilities are required to be arbitrarily small. In the first such result we construct two-sided error pda which recognize a non context-free language with arbitrarily small error.

Theorem 3. *Let* $L = \{a^n b^n c^n \mid n \in \mathbb{N}\}$. *Then*

$$L \in \text{BPCF}^* \setminus \text{CF}.$$

One-sided-error pda recognize $\overline{L}$ also with arbitrarily small error, although $\overline{L}$ is not recognizable by Las Vegas pda even if the "I don't know" answer has arbitrarily large probability smaller than one.

Theorem 4. *Let* $L = \{a^n b^n c^n \mid n \in \mathbb{N}\}$. *Then*

$$\overline{L} \in \text{RandomCF}^* \setminus \text{LVCF}.$$

Our two final separation results limit the power of two-sided error pda's. We begin by showing that although randomization may increase recognition power even beyond nondeterminism, randomization is far too weak to simulate guessing in general.

Theorem 5. *There is a context-free language L with*

$$L \in \text{CF} \setminus \text{BPCF}.$$

In particular, L cannot be recognized by a probabilistic pda with error at most $\frac{1}{2} - c \cdot \frac{\log_2 n}{n}$, *where n is the length of the input and c is a suitably large constant*.

Thus nondeterminism can be even stronger than probabilism with weakly-unbounded-error.

Demanding arbitrarily small error probabilities may result in a severe loss of recognition power, since some languages recognizable by Las Vegas pda now turn out to be too hard. Below the symbols $\$_1$ and $\$_2$ are used as end markers

Theorem 6. *There are deterministic context-free languages* L_1, L_2 *with*

$$L_1 \cup L_2 \in \text{LVCF}_{1/2} \setminus \text{BPCF}^*.$$

Thus two-sided error pda may loose dramatically in recognition power, if error probabilities have to be arbitrarily small.

Observe that two-sided error pda are capable of recognizing any union $L_1 \cup L_2$ of deterministic context-free languages with error probability $\frac{1}{3}$ as follows: if A_1, A_2 are dpda's for L_1 and L_2, respectively, then for input w flip a coin with probability $\frac{1}{2}$ to decide whether to simulate A_1 or A_2 on w. Accept, if the simulated dpda accepts, but reject with probability $\frac{2}{3}$, if the simulated dpda rejects. If $w \notin L_1 \cup L_2$, then our simulating pda P errs with probability $\frac{1}{3}$, if $w \in L_1 \cup L_2$, then P errs with probability at most $\frac{1}{2} \cdot \frac{2}{3} = \frac{1}{3}$. As a consequence of Theorem 7(a) this trivial recipe cannot be improved.

Theorem 7.

(*a*) *There are deterministic context-free languages* L_1, L_2 *with*

$$L_1 \cup L_2 \in \text{LVCF}_{1/2} \setminus \text{BPCF}_{\frac{1}{3} - 2^{-n/8 + c \cdot \log_2 n}},$$

where c is a suitably large constant.

(*b*) *Assume that* L_1, L_2 *are deterministic context-free languages and that the symbol* \$ *does not belong to the alphabets of* L_1 *or* L_2. *If* $L_1 \cap L_2$ *is not context-free, then*

$$L_1 \cup L_2\$ \in \text{LVCF}_{1/2} - \bigcup_{p<1/2} \text{RandomCF}_p.$$

Thus $1/3$ is a sharp threshold when recognizing a union of two deterministic context-free languages by two-sided error pda and $1/2$ is a sharp threshold for one-sided-error pda: a one-sided-error pda reaches recognition probability $1/2$ if it randomly selects one of A_1, A_2 and then simulates it.

To show Theorems 5, 6 and 7(a) we apply methods from communication complexity, but face a severe problem, since a traditional simulation of pda by communication cannot handle the large amount of information stored in the stack. Hence we

have to design new communication models that are powerful enough to be applicable to pda, but also weak enough so that their power can be controlled. The resulting method for proving lower bounds on probabilistic pda's is the main technical contribution of this paper.

This paper is organized as follows. Section 2 is devoted to a study of closure properties for the above probabilistic language classes which are useful in providing inter-class relations. In particular we show Lemma 9, a stronger version of Theorem 7(b).

We give a proof of Theorems 3 and 4 in Section 3 and use these results as well as Theorems 5 and 6 to establish Theorem 2. The deferred proofs of Theorems 5 and 6 as well as the proof of Theorem 7a are given in Sections 4.2 and 4.3, respectively, the non-standard communication model is described in Section 4.1.

2. Some closure and non-closure properties

We say that a Kleene closure L^* is marked, if the words in L end in letters that only appear at the end. We first consider closure properties of Las Vegas languages.

Lemma 8.

(*a*) *Let L_1, L_2 be deterministic context-free languages over an alphabet not containing the symbol* \$. *Then $L_1 \cup L_2\$ \in LVCF_{1/2}$.*
(*b*) *Assume $0 < p < 1$. If $L \in LVCF_p$, then $\bar{L} \in LVCF_p$.*

Proof. **(a)** We describe a Las Vegas pushdown automata P that recognizes $K = L_1 \cup L_2\$$. For input w, P first tosses a fair coin to decide whether to bet on $w \in L_1$ or to bet on $w \in L_2\$$.

Case 1: P bets on $w \in L_1$. P simulates a deterministic pda D_1 for L_1 and accepts if D_1 accepts. Moreover, P rejects w if and only if D_1 rejects w and the last letter of w is different from \$. Finally, if D_1 rejects and the last letter of w is equal to \$, then P answers with a question mark.

Case 2: P bets on $w \in L_2\$$. P simulates a deterministic pda D_2 for L_2 and accepts (resp. rejects), if the last letter is equal to \$ and D_2 has accepted (resp. rejected) in the previous step. Finally, if the last letter is not equal to \$, then P answers with a question mark.

Observe that P does not make any error and outputs a question mark with probability $\frac{1}{2}$.

(b) The argument is analogous to the case of deterministic pda's. □

We show that the probability of $\frac{1}{2}$ for a committing answer of a Las Vegas pda, i.e., an accepting or rejecting answer, cannot be improved for a rather large class of language pairs. Analogously, the probability of a correct answer of a probabilistic pda with one-sided error cannot be improved either.

For languages L_1 and L_2 define the new language

$$(L_1, L_2) = \{\, u\#v \mid u \in L_1 \text{ and } u \cdot v \in L_2 \,\},$$

where we assume that the new letter # does not belong to the alphabets of L_1 or L_2. Theorem 7(b) is an immediate consequence of the following observation.

Lemma 9. *Assume that L_1, L_2 are deterministic context-free languages and that the symbol* \$ *does not belong to the alphabets of L_1 or L_2.*

If $L_1 \cap L_2$ or (L_1, L_2) is not context-free, then

$$L_1 \cup L_2\$ \in \text{LVCF}_{1/2} - \bigcup_{p<1/2} \text{RandomCF}_p.$$

Proof. Since L_1, L_2 are deterministic context-free languages, we know by part (a) of Lemma 8 that $L_1 \cup L_2\$$ belongs to $\text{LVCF}_{1/2}$. Thus we have to show that $L_1 \cup L_2\$$ does not belong to $\bigcup_{p<1/2} \text{RandomCF}_p$, provided $L_1 \cap L_2$ or (L_1, L_2) is not context-free. Assume otherwise and let Q be a one-sided-error pda which recognizes $L_1 \cup L_2\$$ with error probability less than $\frac{1}{2}$.

Case 1: $L_1 \cap L_2$ is not context-free. Let u be an arbitrary word and assume that $u \cdot v_1 \in L_1$ as well as $u \cdot v_2 \in L_2\$$. Then there must be a Q-computation on u which is extendable to an accepting computation on $u \cdot v_1$ as well as to an accepting computation on $u \cdot v_2\$$, since otherwise Q has error probability at least $\frac{1}{2}$.

Thus $L_1 \cap L_2$ can be recognized by a nondeterministic pda Q_1 which simulates Q on input u and accepts u if and only if Q accepts u and then subsequently $u \cdot \$$. Thus $L_1 \cap L_2$ is context-free contradicting our assumption.

Case 2: (L_1, L_2) is not context-free. We accept (L_1, L_2) by a nondeterministic pda Q_2 as follows. Q_2 simulates Q until Q reads the symbol #. Then Q_2 checks whether Q is in an accepting state and if no \$ was previously read. If this is the case, then Q_2 continues its simulation and otherwise enters a terminally rejecting state. From now on however Q_2 accepts only if Q would have accepted *after* reading the dollar symbol. □

Thus $\frac{1}{2}$ is a sharp error threshold for a large class of languages $L_1 \cup L_2\$$. For a first example set $L_1 = \{a^n b^n c^m \mid n, m \in \mathbb{N}\}$ and $L_2 = \{a^m b^n c^n \mid n, m \in \mathbb{N}\}$. Then

$$\{a^n b^n c^m \mid n, m \in \mathbb{N}\} \cup \{a^m b^n c^n\$ \mid n, m \in \mathbb{N}\} \in \text{LVCF}_{1/2} - \bigcup_{p<1/2} \text{RandomCF}_p,$$

since $L_1 \cap L_2$ is not context-free. As a second example consider $K_1 = \{a^n b^n \mid n \in \mathbb{N}\}$ and $K_1 = \{a^n b^{2n} \mid n \in \mathbb{N}\}$. Obviously $(K_1, K_2) = \{a^n b^n \# b^n \mid n \in \mathbb{N}\}$ is not context-free and $\frac{1}{2}$ is a sharp threshold also for $K_1 \cup K_2\$$, since

$$\{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2n}\$ \mid n \in \mathbb{N}\} \in \text{LVCF}_{1/2} - \bigcup_{p<1/2} \text{RandomCF}_p.$$

Neither LVCF nor RandomCF turn out to be closed under the marked Kleene closure. To simplify the argumentation we need the following fact on "predicting machines".

Fact 10. *([5, p. 240])*. For every dpda $D = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ there is an equivalent dpda $D' = (Q, \Sigma, \Gamma', \delta', q_0, Z_0', F)$ with the following properties:

(a) $\Gamma' = \Gamma \times \mathcal{P}(Q)$, where $\mathcal{P}(Q)$ is the power set of Q.
(b) If a computation of D on some input u has reached state q and created the stack $S = (Z_1, \ldots, Z_k)$, then D' on input u has reached state q as well and has created the stack $S' = (Z_1, P_1) \cdots (Z_k, P_k)$ with

$$P_1 = \{q \in Q \mid \text{there is a string } v \in \Sigma^* \text{ and an accepting computation of } D \text{ on } v \text{ with initial state } q \text{ and stack contents } S\}.$$

Thus D' works exactly as D, but remembers additionally in its stack symbols for which states an accepting computation is possible, given its current stack contents. Observe that we may also assume that probabilistic pda's are equipped with this prediction mechanism, if transition probabilities for the predicting machine are defined exactly as for the original pda.

Lemma 11. *Let L be a language over an alphabet not containing the letter #.*

(a) If $(L\#)^* \in$ LVCF, then $L \in$ LVCF*.
(b) If $(L\#)^* \in$ RandomCF, then $L \in$ RandomCF*.
(c) Neither LVCF *nor* RandomCF *are closed under the marked Kleene closure.*

Proof. (a) We may assume that P is a Las-Vegas pda which recognizes $(L\#)^*$ with probability $\delta > 0$. (W.l.o.g. we may also assume that there are words $x^{(n)} \in (L\#)^*$ such that the sequence of acceptance probabilities of $x^{(n)}$ converges to limit δ.) We assume that P is a predicting machine and hence, at any time P knows if it is possible to eventually enter an accepting state. In particular we may additionally assume that, if P accepts $u\#v\#$ in some computation, then it accepts $u\#$ in that computation as well.

For a given ε $(0 < \varepsilon < 1)$ choose words $w_1, \ldots, w_k \in L$ such that

$$\text{prob}[\ P \text{ accepts } x\ ] \leq \frac{\delta}{1 - \varepsilon/2},$$

where $x = w_1\# \cdots \# w_k\#$. Let $w \in L$ be arbitrary. We get

$$\begin{aligned} \text{prob}[\ P \text{ accepts } x \cdot w\# \mid P \text{ accepts } x\ ] &= \frac{\text{prob}[\ P \text{ accepts } x \text{ and } x \cdot w\#\ ]}{\text{prob}[\ P \text{ accepts } x\ ]} \\ &= \frac{\text{prob}[\ P \text{ accepts } x \cdot w\#\ ]}{\text{prob}[\ P \text{ accepts } x\ ]} \\ &\geq \frac{\delta}{\delta/(1-\varepsilon/2)} = 1 - \varepsilon/2. \end{aligned}$$

This observation suggests the following Las Vegas pda P'. P' simulates P on the "virtual" input $w_1\# \cdots \# w_k\#$ until an accepting state is reached. (In order not to get caught in an infinite computation P' will count the number of steps per try in its states and stop the try, if a predetermined threshold is reached.) If an accepting state is eventually reached, then P' continues the simulation of P by reading the "real" input w. By supplying a sufficiently large threshold we can guarantee that P' accepts any $w \in L$ with probability at least $1 - \varepsilon$.

Here δ is simply a lower bound on the acceptance property. You cannot guarantee the existence of x for which the acceptance probability is at most $\delta/(1 - \epsilon/2)$.

(b) follows analogously. (c) is a consequence of (a) and (b), since amplified Las Vegas pda's (resp. amplified pda's with one-sided error) are weaker than Las Vegas pda's (resp. pda's with one-sided error) by Lemma 9. □

We next summarize some non-closure properties for LVCF.

Lemma 12.

(*a*) LVCF *is not closed under finite union with languages from* DCF.
(*b*) LVCF *is not closed under concatenation after a regular language*. *Moreover,* LVCF *is not closed under marked Kleene closure*.

Proof. (a) If LVCF would be closed under finite union with languages from DCF, then it would also be closed under finite intersection with deterministic context-free languages. This is obviously false as for instance $\{ a^n \cdot b^n \cdot a^m \mid n, m \in \mathbb{N} \} \cap \{ a^m \cdot b^n \cdot a^n \mid n, m \in \mathbb{N} \}$ is not context-free.

(b) We use the standard construction to show non-closure under concatenation after the regular language $\$^*$. Let $L_1, L_2 \in$ DCF be languages over the alphabet Σ such that $\overline{L_1} \cap \overline{L_2}$ is not context-free. Observe that $K = \$L_1 \cup L_2$ is a deterministic context-free language. If the concatenation $\$^* \cdot K$ belongs to LVCF, then so does $\$L_1 \cup \$L_2$. But then obviously $L_1 \cup L_2 \in$ LVCF and we obtain a contradiction, since LVCF is closed under complementation.

LVCF is not closed under marked Kleene closure as a consequence of Lemma 11(c). □

We now consider closure properties of RandomCF under finite union, marked Kleene closure and complemtation.

Lemma 13.

(*a*) RandomCF *is closed under finite union with languages from* DCF.
(*b*) RandomCF *is neither closed under marked Kleene closure nor under complementation*.

Proof. (a) Closure under finite union is obvious, since the union of k deterministic context-free languages can be recognized with probability at least $\frac{1}{k}$.

(b) But RandomCF is not closed under the marked Kleene closure as a consequence of Lemma 11(c).

Now assume that RandomCF is closed under complementation. Observe that RandomCF $\subseteq$ CF, since a one-sided-error pda does not err when accepting. Since RandomCF is closed under finite union with deterministic context-free languages, RandomCF is also closed under finite intersection with deterministic context-free languages. Thus CF would be closed under finite intersection of deterministic context-free languages as well, a contradiction. □

3. Separation results

We begin by noting that Las Vegas computations are a special form of randomized computations with one-sided error.

Proposition 14.

(*a*) *For every* ε $(0 \leq \varepsilon \leq 1)$, $\text{LVCF}_\varepsilon \subseteq \text{RandomCF}_\varepsilon$.
(*b*) LVCF $\subseteq$ RandomCF *and* $\text{LVCF}^* \subseteq \text{RandomCF}^*$.

Proof. Since (b) is a consequence of (a) it suffices to show (a). Let P be a Las Vegas PDA with recognition probability $1 - \varepsilon$ so that $L(P)$ belongs to LVCF_ε. It suffices to construct an equivalent pda P' with one-sided error at most ε: P' simulates P and gives the same output, provided P commits itself. If P is non-committal, which happens with probability at most ε, then P' rejects. □

Our goal is to show Theorems 3 and 4. Observe that as consequence of Theorem 4 both inclusions in part (b) of Proposition 14 are proper. We begin with Theorem 3 and show

$$L \in \text{BPCF}^* \setminus \text{CF}$$

for $L = \{a^n b^n c^n \mid n \in \mathbb{N}\}$.

Proof of Theorem 3. We construct a one-sided-error pda P_N for L which randomly decides to simulate one of a collection of deterministic one-counter automata $(Q_x \mid 1 \leq x \leq N)$. For an input word $w = a^i b^j c^k$ the automaton Q_x determines $\alpha_{i,j,k}(x) = i + j \cdot x - k \cdot (x + 1)$ through appropriate counter movements and accepts w iff $\alpha_{i,j,k}(x) = 0$. Any input w which does not belong to $a^* b^* c^*$ is rejected.

The pda P_N picks $x \in \{1, \ldots, N\}$ uniformly at random and simulates Q_x. If w belongs to L, then $\alpha_{i,j,k}(x) = 0$ and P_N does not err for inputs belonging to L. Now assume that $w = a^i b^j c^k$ does not belong to L. Observe that

$$\alpha_{i,j,k}(x) = i + j \cdot x - k \cdot (x + 1) = (i - k) + x \cdot (j - k)$$

and the condition $\alpha_{i,j,k}(x) = 0$ is equivalent to $(i - k) = x \cdot (k - j)$. Thus there is at most one choice for x with $\alpha_{i,j,k}(x) = 0$ and P_N recognizes L with error probability at most $\frac{1}{N}$. □

Proof of Theorem 4. Observe that $\bar{L}$ belongs to RandomCF*, since L was recognized without erring on words in L. Moreover $L \notin$ LVCF, since L is not context-free. But since LVCF is closed under complementation, $\bar{L}$ also does not belong to LVCF. □

We are now ready to verify the separation results claimed in Theorem 2. All separation results follow from one of Theorems 3-6. Since Theorems 5 and 6 are quite non-trivial, we also give alternate arguments whenever possible.

Proof of Theorem 2. (a) DCF is a proper subset of LVCF as a consequence of Theorem 6. An elementary alternate argument applies Lemma 9 to $L = \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a^n b^{2n} \$ \mid n \in \mathbb{N}\}$. We get

$$L \in \text{LVCF}_{1/2} - \bigcup_{p<1/2} \text{RandomCF}_p \tag{1}$$

By Proposition 14, LVCF $\subseteq$ RandomCF. This inclusion is proper as a consequence of Theorem 4.

RandomCF is a proper subset of CF by Theorem 5. An elementary alternate argument observes that CF, but not RandomCF is closed under marked Kleene closure (Lemma 13(b)). Finally RandomCF is a proper subset of BPCF by Theorem 3, since RandomCF is contained in CF.

(b) CF and BPCF are incomparable as a consequence of Theorem 3 and Theorem 5.

(c) LVCF*, RandomCF* and BPCF* are proper subsets of LVCF, RandomCF and BPCF, respectively, by Theorem 6. Applying (1) provides an alternative argument for LVCF* and RandomCF*.

(d) By Proposition 14, LVCF$^* \subseteq$ RandomCF* and this inclusion is proper by Theorem 4. Finally RandomCF* is a proper subset of BPCF* by Theorem 3.

(e) RandomCF* and LVCF are incomparable, since Theorem 4 shows that RandomCF* is not a subset of LVCF and LVCF is not a subset of RandomCF* by Theorem 6 (or by applying (1)).

We show next that BPCF* is incomparable with all three classes LVCF, RandomCF and CF. Firstly, the recognition power of BPCF* is limited by Theorem 6, which shows that none of the classes is contained in BPCF*. But BPCF* contains a non-context-free language according to Theorem 3. □

4. Two-sided error

In this section we introduce a non-standard model of communication and describe the proofs of Theorems 5, 6 and 7(a).

The class of languages recognizable by probabilistic pda's with bounded-error seems to have lost any resemblance of the pumping-property, since for instance the language $\{a^n b^n c^n \mid n \in \mathbb{N}\}$ is recognizable with even arbitrarily small error. Thus structural reasons as limits on the computing power seem unlikely. Therefore we try to apply methods from communication complexity, but are immediately confronted with the problem of dealing with a potentially large stack which may encode the entire input seen so far. Hence we develop the two-trial communication model, a non-standard model of communication which is tailor-made to handle pda.

4.1. Two-trial communication

A probabilistic pda P on input w generates a computation tree T_w which lists all computations of P on w. Any path from the root of T_w to a leaf is called a deterministic computation of P on w.

Definition 15. Let P be a probabilistic pda and let C be a deterministic computation of P on input w. We define $\text{stack}_C(w)$ to equal the contents of the stack after reading w according to C and just before reading the next input letter, $\text{height}_C(w)$ denotes the height of $\text{stack}_C(w)$.

We say that C compresses u_2 relative to the partition (u_1, u_2, v_1) of input $u_1u_2v_1$ iff the lowest stack height h when reading u_2 is at least as large as the lowest stack height when reading v_1. We additionally demand that $h \leq \text{stack}_C(u_1)$ and $h \leq \text{stack}_C(u_1u_2)$.

We introduce the two-trial communication model to simulate a probabilistic pda P on input w. Two processors A and B participate. The input w is arbitrarily partitioned into four substrings $w = u_1u_2v_1v_2$: processor A receives the pair (u_1, u_2) and processor B receives the (v_1, v_2).

When reading v_1, the deterministic computation C has the option to compress u_2. Therefore we simulate P by a probabilistic protocol in two trials, assuming first that C does not compress u_2 and then assuming that C does compress u_2. The protocol assumes public random bits and decides whether or not to accept w.

The following definition formalizes two-trial communication. (A deterministic computation of a probabilistic protocol $\mathcal{P}$ on input w corresponds to a path from the root of the protocol tree for $\mathcal{P}$ on input w to a leaf.)

Definition 16. Let $c : \mathbb{N} \to \mathbb{N}$ be a given function. A two-trial randomized communication protocol $\mathcal{P}$ with communication at most $c(n)$ is defined as follows.

(a) Processor A receives (u_1, u_2) and processor B receives (v_1, v_2) as input. We set $u = u_1u_2$, $v = v_1v_2$ and $w = uv$. We assume public random bits throughout.
(b) In trial 1 A sends u_2 and an additional message of length at most $c(|w|)$. Either B sends a question mark or B commits and replies by sending v_2 and an additional message of length at most $c(|w|)$. B's decision to commit does not depend on v_2.
(c) In trial 2 B sends v_1. Either A sends a question mark or A commits and replies by sending u_1 and an additional message of length at most $c(|w|)$. A's commitment decision is based only on u_2, v_1 and a string s_{u_1,u_2}. The string s_{u_1,u_2} has length $O(\log_2(|u|))$ and depends only on u_1 and u_2.
(d) For every deterministic computation of $\mathcal{P}$ on input w exactly one of the two trials commits and one processor has to determine the output.

Observe that we do not charge for exchanging u_2, v_2 in trial 1, resp. exchanging u_1, v_1 in trial 2 and charge only for the additional information. The decision to commit has become a powerful new feature of the new model and therefore we demand that commitment can be determined with restricted input access.

Next we define recognition of languages. We require the error probability for every input w and for every partition of w to be small. A question mark is not counted as an error, but property (d) demands that for every deterministic computation exactly one trial leads to commitment.

Definition 17. Let $L \subseteq \Sigma^*$ be a language and let $\mathcal{P}$ be a two-trial randomized communication protocol. For an input w and a partition $p = (u_1, u_2, v_1, v_2)$ with $w = u_1u_2v_1v_2$ we define the error probability of w relative to p to be

$$\varepsilon_p(w) = t_p^1(w) \cdot \varepsilon_p^1(w) + t_p^2(w) \cdot \varepsilon_p^2(w),$$

where $\varepsilon_p^i(w)$ is the error probability for w in trial i and $t_p^i(w)$ is the probability that the processors commit in trial i on input w relative to partition p. (Hence $\varepsilon_P(w)$ only counts a misclassification as an error and disregards question marks.)

We say that $\mathcal{P}$ recognizes L with error probability at most ε iff $\varepsilon_p(w) \leq \varepsilon$ for every input w and for every partition p of w.

Observe that $t_p^i(w) \cdot \varepsilon_p^i(w)$ is the error probability of $\mathcal{P}$ conditioned on committing in trial i. (However $\varepsilon_p^i(w)$ is the unconditional probability of erring in phase i, since any computation on input w performs both trials.) By property (d) of Definition 16, $\varepsilon_p(w) = t_p^1(w) \cdot \varepsilon_p^1(w) + t_p^2(w) \cdot \varepsilon_p^2(w)$ is indeed the error probability of $\mathcal{P}$ on input w, since exactly one of the two trials commits in any deterministic computation of $\mathcal{P}$.

We now show how to simulate the probabilistic pda P with the two-trial communication model. Our goal is to exchange as little additional information as possible.

In **trial 1** the simulation will be successful, if C does not compress u_2 relative to the partition (u_1, u_2, v_1). In particular, let h be the lowest stack height when reading u_2 and let T_1 be the last time during the processing of u_2 when the stack has height h. (At time T the automaton has just performed the Tth instruction.) A sends

1. a pointer to the first unused random bit at time T_1,
2. the state and the topmost stack symbol at time T_1,
3. u_2 and a pointer to the first unread input symbols of u_2 at time T_1.

Processor B will be able to simulate P, beginning at time T_1, as long as the stack height is at least as large as h. If the stack height decreases to $h - 1$ when reading v_1, then B stops the trial by sending a question mark. Otherwise B commits and we observe that B's commitment decision does not depend on v_2. If the stack height reaches height $h - 1$ at time T_2, then B sends

4. a pointer to the first unused random bit at time T_2,
5. the current state at time T_2,
6. v_2 and a pointer to the first unread input symbol of v_2 at time T_2.

and processor A can finish the simulation. Thus A sends u_2, followed by B who sends v_2. Moreover both processors exchange $O(\log_2(|w|))$ additional bits. The simulation is successful, provided P does not compress u_2 relative to (u_1, u_2, v_1). Also remember that B can determine whether this trial is successful without consulting v_2.

But trial 1 may fail, if C does compress u_2 relative to the partition (u_1, u_2, v_1). Therefore **trial 2** assumes compression. Processor B begins by sending v_1 and A replies with a question mark if u_2 is not compressed. Otherwise A commits and continues the simulation which results in compressing u_2. Assume that h is the lowest stack height when reading v_1 and that height h is reached at time T for the last time during the processing of v_1. Observe that $h \leq \text{height}_C(u_1)$, since u_2 is compressed. A sends

1. a pointer to the first unused random bit at time T,
2. the state at time T and the height h,
3. u_1 and a pointer to the first unread input symbols of v_1 at time T.

B first determines $\text{stack}_C(u_1)$ by simulating C on u_1 and then determines the stack at time T, which consists of the h bottommost stack elements of $\text{stack}_C(u_1)$. Then B finishes the computation by simulating C from time T onwards with the help of the remaining information. Observe that, disregarding $O(\log_2(|w|))$ bits of additional information, B sends v_1, followed by A who sends u_1. The simulation is successful, provided C compresses u_2 relative to (u_1, u_2, v_1).

Moreover A's decision to commit can be based only on the lowest stack height h' when reading u_2, the top portion of the stack after reading u_1u_2 (i.e., the stack elements with height larger than h'), the state after reading u_1u_2 and the string v_1. To determine the top portion of the stack, A just has to know the state and stack element after visiting height h' at time t for the last time, the first unread position of u_2 and the first unused random bit at time t as well as u_2. Thus knowledge of u_2, v_1 and additional information on u_1 and u_2 of logarithmic length is sufficient.

We summarize our above simulation.

Lemma 18. *Let P be a probabilistic pda. Assume that P recognizes the language L with error probability at most ε. Then L can be recognized in the two-trial model with communication $O(\log_2 n)$ for input length n and error probability at most ε.*

The Lemma also holds for pda's and dpda's. However the resulting lower bounds will not always be best possible. For instance $\{a^n b^n c^n \mid n \geq 0\}$ can be recognized in the deterministic two-trial model with communication $O(\log_2 n)$, since A can encode its entire input with logarithmically many bits.

To observe the power of randomized two-trial protocols consider the language

$$\text{ND} = \{u\#v \mid u, v \in \{0,1\}^* \text{ and there is } i \text{ with } u_i = v_i = 1\}$$

of non-disjointness. ND can probably not be recognized with bounded-error by a probabilistic pushdown automata, however the following two-trial protocol recognizes ND with error at most $\frac{1}{3}$ without any (charged) communication: the processors commit with probability $\frac{1}{2}$. If a common element is determined after exchanging u_1, v_1 (resp. u_2, v_2), then accept with probability 1 and otherwise accept with probability $\frac{1}{3}$. Hence the error is $\frac{1}{3}$ for disjoint sets and otherwise the error is at most $\frac{1}{2} \cdot \frac{2}{3} = \frac{1}{3}$.

ND is the prime example for separating probabilism and nondeterminism within conventional two-party communication [12,16]. Thus a separation of probabilism and nondeterminism for pds's remains non-trivial.

4.2. Proof of Theorem 5

Our goal is to reduce the two-trial communication problem for IP to a conventional one-way randomized communication problem. We fix a natural number N and consider an arbitrary probabilistic pda P for IP. Set

$$f_P(v_1) = \sum_{u_1u_2 \in \Sigma^{2N}} \text{prob}[P \text{ compresses } u_2 \text{ for partition } (u_1, u_2, v_1)].$$

We show in Proposition 19 that a string v_1 can be constructed such that the probability of compression w.r.t. (u_1, u_2, v_1) is, on the average, almost as high as the probability of compression w.r.t. (u_1, u_2, v_1v_2) for strings $v_2 \in \Sigma^{2N}$. (Observe that the probability of compression does not decrease when appending suffices.)

We make v_1 known to both processors in a simulating two-trial protocol. If processor A receives (u_1, u_2, v_1), then A can determine whether trial 1 fails. If so, then A, already knowing v_1, sends u_1 and a small amount of information enabling B to continue the simulation by itself. If trial 1 succeeds, then A sends u_2 and again additional information for B to continue. But this time B will, with high probability, not have to respond, since trial 1 will remain successful with high probability for suffix v_1v_2. Thus the two-trial communication model "almost" turns one-way, if both processors know v_1. In particular, the issue of commitment has disappeared. We begin with the construction of $v = v_1$.

Proposition 19. *Let $\Delta \in \mathbb{R}$, $\Delta > 0$, be given. Then there is a string $v \in \Sigma^*$ of length at most $2N \cdot \frac{|\Sigma|^{2N}}{\Delta}$ such that $f_P(vw) \leq \Delta + f_P(v)$ for all $w \in \Sigma^{2N}$.*

Proof. We obtain $f_P(v) \leq f_P(vw)$, since the probability of compression does not decrease when appending suffices. We now construct a string v iteratively as follows:

(1) Set $i = 0$ and $v^0 = \lambda$, where λ is the empty string.
(2) If there is a string $v' \in \Sigma^{2N}$ with $f_P(v^i v') - f_P(v^i) \geq \Delta$, then set $v^{i+1} = v^i v'$, $i = i + 1$ and go to (2). Otherwise stop and output $v = v^i$.

Observe that there are at most $\frac{|\Sigma|^{2N}}{\Delta}$ iterations, since the "f-score" increases by at least Δ in each iteration and since the maximal f-score is $|\Sigma|^{2N}$. □

We fix Δ and obtain a string v with the properties stated in Proposition 19. For an arbitrary language L define

$$L_{N,v} = \{\ (u, w) \mid |u| = |w| = 2N \text{ and } uvw \in L\ \}.$$

We now utilize that the two-trial protocol of Lemma 18 collapses to a conventional one-way randomized protocol with public randomness and small expected error.

Lemma 20. *Fix the parameters $N \in \mathbb{N}$ and $\Delta \in \mathbb{R}$, $\Delta > 0$. If L is recognized by a probabilistic pda P with error probability at most ε, then $L_{N,v}$ can be recognized by a conventional one-way randomized communication protocol in the following sense*:

(1) *String $u \in \{0,1\}^{2N}$ is assigned to processor A and string $w \in \{0,1\}^{2N}$ is assigned to processor B. Both processors know v.*
(2) *For each u and for each w, the communication protocol achieves error probability at most $\varepsilon + p_{u,w}$ on input (u,w), where*

$$\sum_{u \in \Sigma^{2N}} \sum_{w \in \Sigma^{2N}} p_{u,w} \leq \Delta \cdot |\Sigma|^{2N}.$$

(3) *Processor A sends a message of $O(\log_2(|u|+|v|))$ bits and additionally either u_1 or u_2 is sent. u_1 is the prefix, u_2 is the suffix of $u = u_1u_2$ of length N each.*

Proof. Let u be the input of processor A and w the input of processor B. Let $p_{u,w}$ be the probability that P compresses u_2 relative to (u_1, u_2, vw), but not relative to (u_1, u_2, v). Since v is chosen to satisfy Proposition 19 we have

$$\sum_{u \in \Sigma^{2N}} p_{u,w} = f_P(vw) - f_P(v) \leq \Delta$$

for all $w \in \Sigma^{2N}$. We now simulate P on uvw along the lines of Lemma 18, however this time we only use conventional one-way communcation.

Processor A simulates a computation C of P on input uv. If the computation C does not compress u_2 relative to (u_1, u_2, v), then A behaves exactly as in trial 1 and sends u_2 and $O(\log_2(|u|+|v|))$ additional bits. Now processor B will be able to reconstruct the relevant top portion of the stack obtained by P after reading uv and to continue the simulation as long as the top portion is not emptied. If the top portion is emptied, then B accepts all inputs from this point on. (Observe that this happens with probability at most $p_{u,w}$.)

If the computation C compresses u_2 relative to (u_1, u_2, v), then processor A behaves exactly as in trial 2 and sends u_1 and $O(\log_2(|u|+|v|))$ additional bits. Now processor B can finish the simulation without introducing an additional error. All in all the additional error is bounded by

$$\sum_{u \in \Sigma^{2N}} \sum_{w \in \Sigma^{2N}} p_{u,w} \leq \Delta \cdot |\Sigma|^{2N}$$

and this was to be shown. □

We choose the language

$$\mathrm{IP} = \left\{ uv^{\mathrm{reverse}} \in \{0,1\}^* \mid |u| = |v| \text{ and } \sum_{i=1}^{|u|} u_i \cdot v_i \equiv 1 \bmod 2 \right\}$$

to separate BPCF and CF. We set $\mathrm{IP}_N = \{ uv^{\mathrm{reverse}} \in \mathrm{IP} \mid |u| = |v| = N \}$ and observe that either $\mathrm{IP}_{N,v}$ equals IP_{2N} or it equals the complement of IP_{2N}. Hence, if we assume that IP can be recognized by a probabilistic pushdown P with error probability ε, then we obtain a one-way randomized communication protocol that "almost" recognizes IP_{2N} with error probability "close" to ε.

We set $\delta = \frac{1}{2} - \varepsilon$ and $\Delta = \frac{\delta}{2} \cdot 2^{2N}$. The randomized protocol induced by P introduces an additional total error of at most $\Delta \cdot 2^{2N}$ and hence the total error is at most

$$\varepsilon \cdot 2^{4N} + \Delta \cdot 2^{2N} = \left(\varepsilon + \frac{\delta}{2}\right) \cdot 2^{4N} = \left(\frac{1}{2} - \delta + \frac{\delta}{2}\right) \cdot 2^{4N} = \left(\frac{1}{2} - \frac{\delta}{2}\right) \cdot 2^{4N}.$$

The probabilistic protocol $\mathcal{P}$ uses public random bits as a consequence of Definition 16. Hence we may view $\mathcal{P}$ as a probability distribution over all its deterministic protocols. If π_D is the probability of the deterministic protocol D, then, for any input x,

$$\mathrm{prob}[\mathcal{P} \text{ accepts } x] = \sum_{D, D \text{ accepts } x} \pi_D$$

follows. Hence, by an averaging argument, we obtain a deterministic protocol $\mathcal{D}$ with error at most $\frac{1}{2} - \frac{\delta}{2}$ under the uniform distribution.

Next we derive a lower bound for such protocols.

4.2.1. The discrepancy method

Let X and Y be finite sets and let $L \subseteq X \times Y$ be a language. We say that R is a rectangle, if $R = X' \times Y'$ for subsets $X' \subseteq X$ and $Y' \subseteq Y$. The discrepancy $D_\mu(R, L)$ of L with respect to a rectangle R and a distribution μ is defined as

$$D_\mu(R,L) = \left| \sum_{(x,y)\in R \text{ and } (x,y)\notin L} \mu(x,y) - \sum_{(x,y)\in R \text{ and } (x,y)\in L} \mu(x,y) \right|.$$

Languages with small discrepancy for all rectangles force conventional deterministic protocols with small error to exchange correspondingly many bits, since large rectangles introduce too many errors.

Fact 21. Let D be a deterministic protocol for a language L with error at most $\frac{1}{2} - \frac{\delta}{2}$ under distribution μ.

(a) If D communicates at most c bits, then D partitions $X \times Y$ into at most 2^c rectangles R_i $(1 \le i \le 2^c)$ such that

$$\delta \le \sum_{i\le 2^c} D_\mu(R_i, L)$$

holds.

(b) Let $\mathrm{IP}_{2N} = \{(u,v) : uv \in \mathrm{IP} : |u| = |v| = 2N\}$ and $X = Y = \{0,1\}^{2N}$. The discrepancy $D_{\text{uniform}}(X' \times Y', \mathrm{IP}_{2N})$ for the uniform distribution is at most

$$\frac{\sqrt{|X'| \cdot |Y'|} \cdot 2^N}{2^{4N}} \le 2^{-N}.$$

Part (a) is Proposition 3.28 in [14]. Part (b) is shown in Example 3.29 of [14].

In the previous section we have derived a deterministic protocol $\mathcal{D}$ for IP_{2N} with error probability $\frac{1}{2} - \frac{\delta}{2}$ under the uniform distribution. The one-way protocol $\mathcal{D}$ either sends u_1 or u_2 as well as $b = O(\log_2(N + |v|))$ bits of additional information. Thus the rectangles $X' \times Y'$ produced by its messages satisfy $|X'| \cdot |Y'| \le 2^N \cdot 2^{2N} = 2^{3N}$ and $D_{\text{uniform}}(X' \times Y', \mathrm{IP}_{2N}) \le 2^{3N/2} \cdot 2^N / 2^{4N} = 2^{-3N/2}$ follows with Fact 21(b). We apply Fact 21(a) and obtain

$$\delta \le \sum_{i\le 2^c} D_{\text{uniform}}(R_i, L) \le 2^c \cdot 2^{-3N/2},$$

assuming that $\mathcal{D}$ exchanges at most c bits and produces the rectangles R_i. As a consequence, $c \ge \log_2(\delta \cdot 2^{3N/2}) = 3N/2 + \log_2 \delta$. But $c = N + b$, where b was the number of bits of additional information, and we have obtained our final conclusion

$$b = \Omega(N/2 + \log_2 \delta).$$

Thus we get

$$\log_2(N + |v|) = \Omega(N/2 + \log_2 \delta). \tag{2}$$

We have $|v| \le 2N \cdot \frac{2^{2N}}{\Delta} = 2N \cdot \frac{2^{2N}}{\frac{\delta}{2}\cdot 2^{2N}} = \frac{4N}{\delta}$ and (2) translates into

$$\log_2 \frac{4N}{\delta} = \Omega(N/2 + \log_2 \delta).$$

Hence we get $\frac{1}{\delta} = 2^{\Omega(N)}$ and the error probability of the probabilistic pda is at least $\varepsilon = \frac{1}{2} - \delta = \frac{1}{2} - 2^{-\Omega(n)}$. To relate the error probability to the length $L = 2N + |v| + 2N$ of the input, we distinguish two cases. If $|v| \le 2^N$, then $\varepsilon = \frac{1}{2} - 2^{-\Omega(n)} = \frac{1}{2} - O\left(\frac{1}{L}\right)$. If $|v| > 2^N$, then $\log |v| \ge N$ and, since $|v| \le \frac{4N}{\delta}$, $\delta = O\left(\frac{\log |v|}{|v|}\right) = O\left(\frac{\log L}{L}\right)$ follows.

4.3. Proof of Theorem 6 and Theorem 7(a)

We begin by proving Theorem 7(a). To show that $\frac{1}{3}$ is a sharp threshold when recognizing a union of two deterministic languages with a two-sided error pda we select the language

$$\mathrm{IP}^2 = \{u_1\#u_2\#v_1\#v_2 \mid (|u_1| = |v_1| \text{ and } u_1v_1 \in \mathrm{IP}) \text{ or } (|u_2| = |v_2| \text{ and } u_2v_2 \in \mathrm{IP})\}.$$

Observe that IP^2 is a union of two deterministic context-free languages.

We now show that our non-standard communication model allows us to sharply bound the error probability when recognizing IP^2. Our analysis concentrates on the input partition $(u_1\#, u_2\#, v_1\#, v_2)$ and we restrict our attention to $\mathrm{IP}^2_N = \{ u_1\#u_2\#v_1\#v_2 \in \mathrm{IP}^2 \mid |u_1| = |v_1| = |u_2| = |v_2| = N \}$. Since the input size equals $4 \cdot N$, it suffices to show that IP^2_N cannot be recognized for sufficiently large N in the two-trial model with communication $O(\log_2 N)$ and error probability at most $\varepsilon = \frac{1}{3} - 2^{-N/2 + c\cdot\log_2 N}$.

Assume otherwise and let $\mathcal{P}$ be a randomized two-trial protocol with error less than ε and communication $O(\log_2 N)$. Our goal is to derive a conventional deterministic protocol $\mathcal{D}$ from $\mathcal{P}$ and then to apply the discrepancy method to $\mathcal{D}$. In particular

we will show that $\mathcal{D}$ has too large error probability with respect to the distribution μ, where μ is the uniform distribution on all inputs (u_1, u_2, v_1, v_2) with $|u_1| = |u_2| = |v_1| = |v_2| = N$ and $u_1 v_1^{\text{reverse}} \notin \text{IP}$ or $u_2 v_2^{\text{reverse}} \notin \text{IP}$. Thus, if (u_1, u_2, v_1, v_2) belongs to IP_N^2, then exactly one of $u_1 v_1^{\text{reverse}} \in \text{IP}$ or $u_2 v_2^{\text{reverse}} \in \text{IP}$ will hold with probability one.

Since $\mathcal{P}$ uses public random bits, we may view $\mathcal{P}$ as a probability distribution over its deterministic protocols. Hence by an averaging argument, analogous the corresponding argument for Theorem 5, there is a deterministic protocol $\mathcal{D}$ with expected error less than ε for distribution μ. Since $\mathcal{P}$ exchanges at most $O(\log_2 N)$ bits, so does $\mathcal{D}$. Remember however that we count only the additional information supplied by the processors.

We begin by investigating a committing *trial-2* message R of $\mathcal{D}$. In this trial-2 message processor B sends v_1 and processor A replies by sending u_1 and additional information I. To specify R we fix the additional information I and require that processor B either accepts or rejects all inputs of R. Observe that R will in general not have the rectangle property, since A's message also depends on v_1. However, if we fix u_1 and v_1, then $R(u_1, v_1) = \{(u_1, u_2, v_1, v_2) \in R \mid u_2, v_2 \in \{0,1\}^N\}$ is a rectangle and thus R is the disjoint union of the rectangles $R(u_1, v_1)$.

We call an input (u, v) *dangerous*, if $u_1 v_1^{\text{reverse}} \notin \text{IP}$ and *harmless* otherwise. (A harmless input is indeed easy, since both processors know u_1 and v_1 and can infer that the combined input belongs to IP_N^2.) For a subset $X \subseteq \{0,1\}^{4N}$ we define $D^+(X)$ and $D^-(X)$ as the set of dangerous inputs of X belonging to IP_N^2, resp. to the complement of IP_N^2; $H(X)$ is the set of harmless inputs of X. Our first goal is to show that messages cannot differentiate between dangerous positive and dangerous negative inputs of $X = R$.

Proposition 22. *For any message R, $| \mu(D^+(R)) - \mu(D^-(R)) | \leq 2^{-N/2}$.*

Proof. We fix u_1 and v_1 with $u_1 v_1^{\text{reverse}} \notin \text{IP}$ and observe that $(u_1, u_2, v_1, v_2) \in R$ belongs to IP_N^2 iff $u_2 v_2^{\text{reverse}}$ belongs to IP_N. Therefore we obtain with Fact 21(b) that

$$D_{\text{uniform}}(R(u_1, v_1), \text{IP}_N) \leq \frac{\sqrt{\text{size of } R(u_1, v_1)} \cdot 2^{N/2}}{2^{2N}} \leq 2^{-N/2}. \tag{3}$$

Remember that message R is the disjoint union of the rectangles $R(u_1, v_1)$. Since we are only interested in dangerous inputs, the claim follows by summing inequality (3) over all pairs (u_1, v_1) with $u_1 v_1^{\text{reverse}} \notin \text{IP}$ and by observing that μ is uniform on dangerous inputs. □

Let $\mathcal{C}$ be the set of all inputs for which a trial-2 message of $\mathcal{D}$ commits. Since $\mathcal{C}$ is a disjoint union of all polynomially many committing trial-2 messages we obtain

$$| \mu(D^+(\mathcal{C})) - \mu(D^-(\mathcal{C})) | \leq \text{poly}(N) \cdot 2^{-N/2}. \tag{4}$$

as a consequence of Proposition 22. Our second goal is to show that the μ-weights of $D^+(\mathcal{C}), D^-(\mathcal{C})$ and $H(\mathcal{C})$ are almost identical. As a first step we show that if we commit in trial 2, then we commit with probability close to $\frac{1}{3}$ for a harmless input.

Proposition 23. $|\frac{\mu(\mathcal{C})}{3} - \mu(H(\mathcal{C})) | \leq \text{poly}(N) \cdot 2^{-N/2}$.

Proof. According to Definition 16, processor A decides its commitment based on its knowledge of the string s_{u_1,u_2}, u_2 and v_1, where the string s_{u_1,u_2} is of length $O(\log_2(|u_1| + |u_2|))$ and only depends on u_1 and u_2.

For the sake of analyzing s_{u_1,u_2} we introduce an artificial "commitment" message from an imaginary processor A_1 with input (u_1, u_2) to an imaginary processor A_2 with input (u_2, v_1). Then A_2 has all the required information do decide whether or not processor A will commit.

Can commitment messages differentiate between $u_1 v_1^{\text{reverse}} \notin \text{IP}$ and $u_1 v_1^{\text{reverse}} \in \text{IP}$? We answer this question with another application of the discrepancy method. First we fix u_2 as well as v_2 and then apply Fact 21(b) to a commitment message. As a result we obtain a discrepancy (of IP_N relative to the uniform distribution) of at most $2^{-N/2}$.

Thus a commitment message cannot differentiate between $u_1 v_1^{\text{reverse}} \notin \text{IP}$ and $u_1 v_1^{\text{reverse}} \in \text{IP}$ and hence cannot differentiate between $(u_1, u_2, v_1, v_2) \in D^+(\mathcal{C}) \cup D^-(\mathcal{C})$ and $(u_1, u_2, v_1, v_2) \in H(\mathcal{C})$, respectively. Since there are polynomially many commitment messages, the overall discrepancy for fixed u_2 and v_2 is at most $\text{poly}(N) \cdot 2^{-N/2}$. Hence, after considering all possible values of u_2 and v_2,

$$\frac{| \, |D^+(\mathcal{C})| + |D^-(\mathcal{C})| - |H(\mathcal{C})| \, |}{2^{4N}} \leq \text{poly}(N) \cdot 2^{-N/2} \tag{5}$$

follows. For a message R let $H^+(R)$ and $H^-(R)$ be the set of harmless inputs of R with $u_2 v_2^{\text{reverse}} \in \text{IP}$ and with $u_2 v_2^{\text{reverse}} \notin \text{IP}$, respectively. Then $| |H^+(R)| - |H^-(R)| | / 2^{2N} \leq 2^{-N/2}$, since the discrepancy of IP_N with respect to $R(u_1, v_1)$ is upper-bounded by $2^{-N/2}$ for every pair (u_1, v_1) with $u_1 v_1^{\text{reverse}} \in \text{IP}$. But we have only polynomially many messages and obtain, after considering all pairs u_1 and v_1,

$$\frac{|\,|H^+(\mathcal{C})| - |H^-(\mathcal{C})|\,|}{2^{4N}} \leq \text{poly}(N)\cdot 2^{-N/2}, \tag{6}$$

respectively, $|\,|H(\mathcal{C})| - 2|H^-(\mathcal{C})|\,| = |\,|H^+(\mathcal{C})| - |H^-(\mathcal{C})|\,| \leq \text{poly}(N)\cdot 2^{-N/2}$. We combine (5) and (6) and obtain

$$\frac{|\,|D^+(\mathcal{C})| + |D^-(\mathcal{C})| - 2|H^-(\mathcal{C})|\,|}{2^{4N}} \leq \text{poly}(N)\cdot 2^{-N/2}.$$

But then $|\ \mu(D^+(\mathcal{C})) + \mu(D^-(\mathcal{C})) - 2\mu(H(\mathcal{C}))\ | \leq \text{poly}(N)\cdot 2^{-N/2}$ and hence

$$\begin{aligned}|\mu(\mathcal{C}) - 3\mu(H(\mathcal{C}))| &= |\mu(D^+(\mathcal{C})) + \mu(D^-(\mathcal{C})) + \mu(H(\mathcal{C})) - 3\mu(H(\mathcal{C}))|\\ &= |\mu(D^+(\mathcal{C})) + \mu(D^-(\mathcal{C})) - 2\mu(H(\mathcal{C}))|\\ &\leq \text{poly}(N)\cdot 2^{-N/2},\end{aligned}$$

which was to be shown. □

Let $(A_i \mid i \leq \text{poly}(N))$ and $(R_i \mid i \leq \text{poly}(N))$ be the sequences of all accepting and rejecting messages of $\mathcal{D}$, respectively. How large is S, the μ-measure of the symmetric difference between the sets of inputs which are correctly, respectively, incorrectly classified by $\mathcal{D}$? If ε_2 is the expected error of trial-2 messages, then the fraction $\mu(\mathcal{C})\cdot(1-\varepsilon_2)$ is classified correctly and the fraction $\mu(\mathcal{C})\cdot\varepsilon_2$ is classified incorrectly. Hence

$$S = \mu(\mathcal{C})\cdot(1-\varepsilon_2-\varepsilon_2). \tag{7}$$

To obtain a hopefully small upper bound on S we introduce

$$S^- = \sum_i |\ \mu(D^+(R_i)) - \mu(D^-(R_i))\ | \text{ and}$$
$$S^+ = \sum_i |\ \mu(D^+(A_i)) + \mu(H(A_i)) - \mu(D^-(A_i))\ |,$$

and observe that $S \leq S^- + S^+$: we may assume w.l.o.g that only dangerous inputs are rejected (i.e., $H(R_i) = \emptyset$ for all i) and hence mistakes can occur only for dangerous inputs. We apply Proposition 22 to S^- and Proposition 23 to S^+ and get, observing that there are only polynomially many messages,

$$S \leq S^- + S^+ \leq \text{poly}(N)\cdot 2^{-N/2} + \frac{\mu(\mathcal{C})}{3}.$$

We can now combine this upper bound of S with the definition (7) of S and obtain

Proposition 24. $\mu(\mathcal{C})\cdot(1-2\cdot\varepsilon_2) \leq \text{poly}(N)\cdot 2^{-N/2} + \frac{\mu(\mathcal{C})}{3}$.

The corresponding claim for trial-1 messages can be shown analogously. Thus, since $\mathcal{D}$ commits itself for each input in exactly one trial due to Definition 16(d), we get $(1-\mu(\mathcal{C}))\cdot(1-2\cdot\varepsilon_1) \leq \text{poly}(N)\cdot 2^{-N/2} + \frac{1-\mu(\mathcal{C})}{3}$, where ε_1 is the expected error of trial-1 messages.

Let ε be the expected error probability of $\mathcal{D}$. Then $\varepsilon = \varepsilon_1\cdot(1-\mu(\mathcal{C})) + \varepsilon_2\cdot\mu(\mathcal{C})$ and we obtain $1-2\cdot\varepsilon \leq \text{poly}(N)\cdot 2^{-N/2} + \frac{1}{3}$ after adding the inequalities for ε_1 and ε_2. The claim $\varepsilon \geq \frac{1}{3} - \text{poly}(N)\cdot 2^{-N/2}$ follows. □

Proof of Theorem 6. Since our goal is to separate BPCF^* from LVCF we work with a *marked* union of deterministic languages. In particular we choose the alphabet $\Sigma = \{0, 1, \#, [1], [2]\}$ and define the language

$$\begin{aligned}\text{IP}^{2,*} = &\{u_1\#u_2\#v_1\#v_2[1] \mid |u_1| = |v_1| \text{ and } u_1v_1 \in \text{IP}\}\cup\\ &\{u_1\#u_2\#v_1\#v_2[2] \mid |u_2| = |v_2| \text{ and } u_2v_2 \in \text{IP}\}.\end{aligned}$$

Thus the end marker indicate in which way the input has to be interpreted. Obviously $\text{IP}^{2,*}$ can be recognized by a Las Vegas pda with recognition probability $\frac{1}{2}$. We proceed as in the argument for part (a) and in particular restrict our attention to trial-2 computations. We have to observe the following changes however.

We demand as before that $|u_1| = |v_1| = |u_2| = |v_2| = N$, but now work with the uniform distribution μ on all inputs in $\{0,1\}^{4N}[1] \cup \{0,1\}^{4N}[1]$, instead of requiring that $u_1v_1 \notin \text{IP}$ or $u_2v_2 \notin \text{IP}$. This time we call an input $(u, v[1])$ harmless and an input $(u, v[2])$ dangerous. Observe that an input is harmless resp. dangerous with probability exactly $\frac{1}{2}$.

Proposition 22 remains correct and hence close to 50% of all dangerous inputs are misclassified. The claim follows, since exactly 50% of all inputs are dangerous. Thus the argumentation is far easier than in part (a). □

5. Open problems

In this paper we did a first step in the investigation of randomized pushdown automata. The following research problems offer further potential progress in the investigation of the nature of randomization within the framework of context-free languages.

1. Is DCF a proper subset of LVCF*? It is not unlikely that the requirement of arbitrarily large commitment probability collapses Las Vegas to determinism.
2. Let co-RandomCF* be the class of all languages L with $\overline{L} \in$ RandomCF*. Is LVCF* a proper subset of RandomCF$^* \cap$ co-RandomCF*?
3. Is there a language L, whose marked Kleene closure is recognizable by a pda with one-sided error $p < 1$, but where L is *not* deterministic context-free?
4. Assume that $L_1 \cup L_2\$$ is recognizable by a pda with one-sided error strictly less than $\frac{1}{2}$. Is $L_1 \cup L_2\$$ deterministic context-free?
5. Does the language $L = \{w \in \{0,1\}^* \mid w = w^{\text{reverse}}\}$ of palindromes belong to BPCF?

Acknowledgments

Many thanks to Jiri Sgall and the referees for helping us to improve the presentation.

References

[1] M. Dietzfelbinger, M. Kutylowski, R. Reischuk, Exact lower bounds for computing Boolean functions on CREW PRAMs, J. Comput. Syst. Sci., 48 (1994) 231–254.
[2] P. Ďuriš, J. Hromkovič, K. Inoue, On the power of nondeterminism and Las Vegas randomization for two-dimensional finite automata, J. Comput. Syst. Sci. 68 (3) (2004) 675–699.
[3] R. Freivalds, Language recognition using probabilistic Turing machines in real time, and automata with a push-down store, Problemi. Peredachi Informacij. 15 (4) (1979) 96–101. (The English translation appears in Probl. Inform. Transm. 15(4) (1979) 319–323.)
[4] R. Freivalds, Projections of languages recognizable by probabilistic and alternating multitape automata, Inf. Process. Lett. 13 (1981) 195–198.
[5] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison-Wesley, 1979.
[6] J. Hromkovič, Communication Complexity and Parallel Computing, Springer, 1997.
[7] J. Hromkovič, Communication protocols — an exemplary study of the power of randomness, in: P. Pardalos, S. Kajasekaran, J. Reif, J. Rolim (Eds.), Handbook on Randomized Computing, Kluwer Publisher, 2001.
[8] J. Hromkovič, G. Schnitger, On the power of Las Vegas II, two-way finite automata, Theor. Comput. Sci. 262 (1) (2001) 1–24.
[9] J. Hromkovič, G. Schnitger, On the power of Las Vegas for one-way communication complexity, OBDD's and finite automata, Inf. Comput. 169 (2) (2001) 284–296.
[10] J. Hromkovič, G. Schnitger, On the power of randomized pushdown automata, in: 5th Int. Conf. Developments in Language Theory, 2001, pp. 262–271.
[11] J. Hromkovič, G. Schnitger, Pushdown automata and multicounter machines: a comparison of computation modes, in: 30th Int. Conf. on Automata, Languages and Programming, 2003, pp. 66–80.
[12] B. Kalyanasundaram, G. Schnitger, The probabilistic communication complexity of set intersection, SIAM J. Discrete Math. 5 (4) (1992) 545–557.
[13] J. Kaneps, D. Geidmanis, R. Freivalds, Tally languages accepted by Monte Carlo pushdown automata, in: RANDOM '97, Lexture Notes in Computer Science 1269, pp. 187–195.
[14] E. Kushilevitz, N. Nisan, Communication Complexity, Cambridge University Press, 1997.
[15] I. Macarie, M. Ogihara, Properties of probabilistic pushdown automata, in: Technical Report TR-554, Dept. of Computer Science, University of Rochester, 1994.
[16] A.A. Razborov, On the distributional complexity of disjointness, Theor. Comput. Sci. 106 (2) (1992) 385–390.
[17] M. Sauerhoff, On the size of randomized OBDDs and read-once branching programs for k-stable functions, Comput. Complex. 10 (2) (2001) 155–178.